

WHAT IS CLAIMED IS:

1. A method comprising:
determining at least one position value of a first object;
determining at least one position value of a second object; and
comparing the position value of the first object with the position value of the second object to determine if the second object is in a queue with the first object.
2. The method of claim 1, said comparing the position value of the first object with the position value of the second object to determine if the second object is in the queue including:
determining a distance between the first object and the second object.
3. The method of claim 2, said determining the distance between the first object and the second object being at substantially a same point in time for the first object and the second object.
4. The method of claim 2, said comparing the position value of the first object with the position value of the second object to determine if the second object is in the queue including:
determining if the determined distance between the first object and the second object is within a predetermined distance threshold.
5. The method of claim 4, said comparing the position value of the first object with the position value of the second object to determine if the second object is in the queue including:
if the determined distance between the first object and the second object is within the predetermined distance threshold, then determining a velocity value of the second object.
6. The method of claim 5, said determining the velocity value of the second object including:
determining a displacement of the second object over a predetermined time period.

7. The method of claim 6, the predetermined time period being a rate at which positions of the first object and second object are sampled.

8. The method of claim 5, said comparing the position value of the first object with the position value of the second object to determine if the second object is in a queue with the first object including:

determining if the determined velocity value of the second object is within a predetermined velocity threshold.

9. The method of claim 8, said comparing the position value of the first object with the position value of the second object to determine if the second object is in a queue with the first object including:

if the determined velocity value of the second object is within the predetermined velocity threshold, then adding the second object to a queue set.

10. The method of claim 9, further comprising:

removing the second object from the queue set if the distance between the first object and the second object exceeds the predetermined distance threshold.

11. The method of claim 9, further comprising

removing the second object from the queue set if the velocity of the second object exceeds the predetermined velocity threshold.

12. The method of claim 1, further comprising:

determining at least a position value of a third object; and

comparing the position value of the third object with at least one of the position value of the first object and the position value of the second object to determine if the third object is in the queue.

13. The method of claim 12, said comparing the position value of the third object with at least one of the position value of the first object and the position value of the second object to determine if the third object is in the queue including:

determining a distance between the third object and at least one of the first object and the second object.

14. The method of claim 13, said determining the distance between the third object and at least one of the first object and the second object being at substantially a same point in time for the third object and at least one of the first object and the second object.

15. The method of claim 13, said comparing the position value of the third object with the position value of at least one of the first object and the second object to determine if the third object is in the queue including:

determining if the determined distance between the third object and at least one of the first object and the second object is within a predetermined distance threshold.

16. The method of claim 15, said comparing the position value of the third object with the position value of at least one of the first object and the second object to determine if the third object is in the queue including:

if the determined distance between the third object and at least one of the first object and second object is within the predetermined distance threshold, then determining a velocity value of the third object.

17. The method of claim 16, said determining the velocity value of the second object including:

determining a displacement of the third object over a predetermined time period.

18. The method of claim 17, the predetermined time period being a rate at which positions of the first object and second object are sampled.

19. The method of claim 16, said comparing the position value of the third object with the position value of at least one of the first object and the second object to determine if the third object is in the queue including:

determining if the determined velocity value of the third object is within a predetermined velocity threshold.

20. The method of claim 19, said comparing the position value of the third object with the position value of at least one of the first object and the second object to determine if the third object is in the queue including:

if the determined velocity value of the second object is within the predetermined velocity threshold, then adding the third object to a queue set.

21. The method of claim 1, said determining at least one position value of the first object including determining a plurality of position values of the first object, and said determining at least one position value of the second object including determining a plurality of position values of the second object.

22. The method of claim 21, further comprising:

determining a time value for each of the plurality of position values of the first object; and

determining a time value for each of the plurality of position values of the second object.

23. The method of claim 22, said comparing the position value of the first object with the position value of the second object to determine if the second object is in the queue including:

determining a distance between the first object and the second object at a plurality of corresponding time values of the first object and the second object.

24. The method of claim 22, said comparing the position value of the first object with the position value of the second object to determine if the second object is in the queue including:

determining a plurality of velocity values of the second object based on at least a subset of the plurality of position values and time values of the second object.

25. The method of claim 1, further comprising:

prior to comparing the position value of the first object and the second object, defining a seed zone within a predetermined distance from a seed location, the seed location corresponding to the beginning of the queue.

26. The method of claim 25, further comprising:
determining if the first object is within the seed zone.

27. The method of claim 26, further comprising:
determining if the first object meets a seed parameter.

28. The method of claim 27, said determining if the first object meets the seed parameter including:

determining if a velocity value of the first object is within a predetermined seed velocity threshold during a predetermined time period.

29. The method of claim 27, said determining if the first object meets the seed parameter including:

determining if the displacement of the first object remains within a threshold displacement during a predetermined time period.

30. The method of claim 27, said determining if the first object meets the seed parameter including:

determining if the first object is within a distance threshold from the seed location during a predetermined time period.

31. The method of claim 27, further comprising:

adding the first object to a queue set if it is determined that the first object meets the predetermined seed parameter.

32. The method of claim 1, further comprising:

adding the second object to a queue set if it is determined that the second object is in the queue.

33. A method, comprising:

determining if a first track associated with a first object meets a predetermined seed parameter, the predetermined seed parameter including at least a position value of the first object;

determining if a second track associated with a second object meets a predetermined queue parameter, the predetermined queue parameter including at least a position value of the second object relative to the position value of the first object.

34. The method of claim 33, wherein said determining if the first track meets the predetermined seed parameter includes:

determining if the first object remains within a fixed area during a predetermined time period.

35. The method of claim 33, wherein said determining if the first track meets the predetermined seed parameter includes:

determining if the first object has a velocity within a threshold velocity during a predetermined time period.

36. The method of claim 33, further comprising:

defining a seed zone within a predetermined distance from a seed location, the seed location corresponding to the beginning of the queue.

37. The method of claim 36, wherein said determining if the first track meets the predetermined seed parameter includes:

determining if a portion of the first track is within the seed zone.

38. The method of claim 36, wherein said determining if the first track meets the predetermined seed parameter includes:

determining if the first object is within a maximum distance from the seed location during a predetermined time period.

39. The method of claim 33, wherein said determining if the second track meets the predetermined queue parameter includes:

determining if the second track remains within a predetermined distance from the first track during a predetermined time period.

40. The method of claim 33, wherein said determining if the second track meets the predetermined queue parameter includes:

determining if the second object has a velocity below a threshold velocity during a predetermined time period.

41. The method of claim 33, further comprising:

adding the first object to a queue set if it is determined that the first track meets the predetermined seed parameter.

42. The method of claim 41, further comprising:

adding the second object to the queue set if it is determined that the second track meets the predetermined queue parameter.

43. The method of claim 33, further comprising:

determining if a third track associated with a third object meets a predetermined third-track queue parameter, the predetermined third-track queue parameter including at least a position value of the third object relative to a position of the second object.

44. The method of claim 43, wherein said determining if the third track meets the predetermined third-track queue parameter includes:

determining if the third track remains within a predetermined distance from the second track during a predetermined time period.

45. The method of claim 44, wherein said determining if the third track meets the predetermined third-track queue parameter includes:

determining if the third object has a velocity that remains within a threshold velocity during a predetermined time period.

46. The method of claim 44, further comprising:

adding the third object to a queue set if it is determined that the third track meets the predetermined third-track queue parameter.

47. An apparatus, comprising:

a processor configured to analyze movement of sensed objects to determine tracks associated with the objects, the processor being further configured to determine at

least one position value of a first object, to determine at least one position value of a second object, and to compare the position value of the first object with the value of the second object to determine if the second object is in a queue with the first object.

48. The apparatus of claim 47, wherein the processor is further configured to determine if the distance between the first object and the second object is within a predetermined distance threshold.

49. The apparatus of claim 47, wherein the processor is further configured to determine if the velocity of the second object is within a predetermined threshold velocity.

50. The apparatus of claim 47, further comprising:
a sensor configured to sense positions of objects within an tracking area.

51. The apparatus of claim 50, wherein the processor is configured to analyze movement of the sensed objects using sensed positions received from the sensor.

52. An apparatus, comprising:
a processor configured to analyze movement of sensed objects to determine tracks associated with the objects, the processor being further configured to determine if a first track associated with a first object meets a predetermined seed parameter, and to determine if a second track associated with a second object meets a predetermined queue parameter.

53. The apparatus of claim 52, wherein the processor is further configured to determine if the first track meets the predetermined seed parameter by analyzing at least one of a position value of the first object and a time value of the first object.

54. The apparatus of claim 52, wherein the processor is further configured to determine if the second track meets the predetermined queue parameter by analyzing at least one of a position value of the second object and a time value of the second object.

55. The apparatus of claim 52, further comprising:
a sensor configured to sense positions of objects within an tracking area.

56. The apparatus of claim 55, wherein the processor is configured to analyze movement of the sensed objects using sensed positions received from the sensor.

57. An apparatus, comprising:

means for determining if a first track associated with a first object meets a predetermined seed parameter, the predetermined seed parameter including at least one of a position value of the first object and a time value of the first object; and

means for determining if a second track associated with a second object meets a predetermined queue parameter, the predetermined queue parameter including at least one of a position value of the second object relative to a position of the first object and a time value of the second object relative to a time value of the first object.

58. A processor-readable medium storing code representing instructions to cause a processor to perform a process, the code comprising code to:

determine a parameter of a first object including at least one position value of the first object;

determine a parameter of a second object including at least one position value of the second object; and

determine if the second object is in a queue with the first object by comparing the parameter of the first object with the parameter of the second object.

59. The processor-readable medium of claim 58, wherein the parameter of the first object includes a time value of the first object, and the parameter of the second object includes a time value of the second object.

60. The processor-readable medium of claim 58, wherein the parameter of the first object includes a velocity value of the first object, and the parameter of the second object includes a velocity value of the second object.